METHOD FOR SELECTING SYNC MARKS
AND SYNC MARK DETECTORS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001]　　This application claims the benefit of U.S. Provisional Application No. 60/306,324, filed July 18, 2001 and U.S. Provisional Application No. 60/327,805, filed October 9, 2001, which are both hereby incorporated by reference.

FIELD OF THE INVENTION

[0002]　　The present invention relates to synchronization marks, and more particularly to the selection and detection of sync marks in a communications channel.

BACKGROUND OF THE INVENTION

[0003]　　Referring now to FIGs. 1 and 2, an incoming bitstream 10 includes synchronization (sync) marks 12 that are positioned after a sync field 14 to identify a starting bit of user data 16. Typically, the sync field 14 includes a repeating pattern such as "1100". A conventional sync mark detector 30 is connected to a front end 32 of a communications channel such as a wireless link, a disk drive or any other communications channel. The front end 32 provides the incoming bitstream 10 to a bit decision circuit 33 such as a Viterbi circuit. An output of the bit decision circuit 33 is coupled to a buffer 34 such as a first in first out (FIFO) buffer. A comparison circuit 36 of the sync mark detector 30 declares

synchronization (or a sync mark found state) if bits stored in the buffer 34 match a predefined sync mark 38 according to a predetermined rule 40. For example for a 3 byte sync mark, the sync mark found state can be declared if 1 of 3 bytes are detected. For a 4 byte sync mark, the sync mark found state can be declared if 2 out of 4 bytes are detected.

[0004]   Referring now to FIG. 1, three different types of sync mark errors may occur during the synchronization process. A sync mark miss is declared if the sync mark detector 30 does not find the predefined sync mark 38 in the incoming bitstream 10. In this case, a retry is required. Misalignment errors occur if the sync mark detector 30 declares the sync mark found state at a wrong position in the incoming bitstream 10. In particular, early alignment 42 refers to the declaration of the sync mark found state before the correct position in the incoming bitstream 10. Late alignment 46 refers to the declaration of the sync mark found state after the correct sync mark position in the incoming bitstream 10. For example when early or late alignment 42 or 46 occur in a disk drive, the bit decision circuit 33 continues operating for the remainder of a current sector and causes an error correction coding (ECC) failure. In this situation, a retry request is generated at the end of the sector.

[0005]   Late alignment 46 occurs when a sync mark miss happens. Bits of the user data 16 after the sync mark 12 match the last several bits of the sync mark 38 so that a sync mark found is declared after the correct position in the incoming bitstream 10. In this situation, the probability of the late alignment 46 is less than the probability of the sync mark miss. Early alignment 42 happens

when the end of the sync field 14 combined with the beginning of the sync mark 12 is similar to the sync mark 38.

[0006]   Referring now to FIG. 3, an example of early alignment is shown. The sync mark includes two 5 bit symbols (11000 and 00011) and a 1-out-of-2 symbol matching rule is used. A single error event, identified by "-", causes the first 5 bits to be the same as the first sync mark symbol and leads to early alignment 42.

SUMMARY OF THE INVENTION

[0007]   A sync mark detector according to the present invention identifies a sync mark in an incoming bitstream of a communications channel. A bit comparing circuit compares the sync mark bit-by-bit to the incoming bitstream if channel irregularities do not exist. A symbol comparing circuit compares the sync mark symbol-by-symbol to the incoming bitstream if channel irregularities do exist.

[0008]   In other features, the channel irregularities cause long consecutive bit errors. A bit decision circuit decodes the incoming bitstream. A first buffer communicates with the bit decision circuit and stores M bits of the incoming bitstream at a desired phase. Alternately, a first post coding circuit communicates with the bit decision circuit. A first buffer communicates with the first post coding circuit and stores M bits of the incoming bitstream at a desired phase. The first post coding circuit performs INRZI or NRZI post coding.

[0009]   In still other features, a second buffer communicates with the bit decision circuit and stores M bits of the incoming bitstream at a desired phase. Alternately, a second post coding circuit communicates with the bit decision circuit. A second buffer communicates with the second post coding circuit and stores M bits of the incoming bitstream at a desired phase. The second post coding circuit performs INRZI or NRZI post coding.

[0010]   In yet other features, the bit comparing circuit implements a bit level matching rule and the symbol comparing circuit implements a symbol level matching rule. The bit comparing circuit generates a sync mark found state when the bit level matching rule is satisfied. The symbol comparing circuit generates a sync mark found state when the symbol level matching rule is satisfied.

[0011]   In yet other features, a longest error event in the sync mark causes x bit errors. A symbol length of the symbols in the incoming bitstream are limited to (x-1). The sync mark includes (4+3s) symbols and the symbol comparing circuit implements a (2+s)-out-of-(4+3s) matching rule for identifying sync marks.

[0012]   Further areas of applicability of the present invention will become apparent from the detailed description provided hereinafter. It should be understood that the detailed description and specific examples, while indicating the preferred embodiment of the invention, are intended for purposes of illustration only and are not intended to limit the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]   The present invention will become more fully understood from the detailed description and the accompanying drawings, wherein:

[0014]   FIG. 1 illustrates an incoming bitstream of a communications channel;

[0015]   FIG. 2 is a functional block diagram of a sync mark detector according to the prior art;

[0016]   FIG. 3 illustrates an exemplary input bitstream with error events and an example of early alignment;

[0017]   FIG. 4 illustrates all-phase and limited-phase detection of the incoming bitstream;

[0018]   FIG. 5 is a table illustrating most likely error events for NRZ and INRZI domains;

[0019]   FIG. 6A is a functional block diagram of a first exemplary sync mark detector according to the present invention;

[0020]   FIG. 6B is a functional block diagram of a second exemplary sync mark detector according to the present invention;

[0021]   FIG. 7 illustrates a sync mark divided into multi-bit symbols and error events that cross symbol boundaries of the sync mark;

[0022]   FIG. 8 illustrates steps performed by the first exemplary sync mark detector of FIG. 6A according to the present invention; and

[0023]   FIG. 9 is a graph illustrating the improved performance of the sync mark detector as compared with a conventional sync mark detector.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0024] The following description of the preferred embodiment(s) is merely exemplary in nature and is in no way intended to limit the invention, its application, or uses.

[0025] To reduce the early alignment probability, an M-bit sync mark should be different than the M previous bits. The M previous bits contain the last K bits of the sync field 14 plus the first M-K bits of the sync mark 12 (where K $\geq$ 1). A minimum Hamming distance d_H of a sync mark is defined as a minimum bit difference between the sync mark and any previous M bits of the incoming bitstream. According to the present invention, the sync mark is divided into several symbols where each symbol contains several consecutive bits. The minimum symbol distance of a sync mark is defined as the minimum symbol difference between the sync mark and any previous M bits.

[0026] In a non return to zero (NRZ) domain, the polarity of a read back signal may be inverted by the front end 32. The sync mark detector 30 compares the incoming bitstream 10 with the sync mark and a bit complement of the sync mark. Therefore, the minimum Hamming distance d_H in the NRZ domain is the minimum bit difference between the sync mark and any M previous bits or their bit complements. The minimum symbol distance in the NRZ domain is defined similarly. When sync mark detection is performed in the NRZI or INRZI domain, the polarity issue does not exist. Therefore the minimum Hamming distance d_H (minimum symbol distance) is the bit (symbol) difference between the sync mark and any M previous bits.

[0027]   Referring now to FIG. 4, when the sync mark detector 30 searches for the sync mark 12, the sync mark detector 30 can start searching at any bit position. Alternately, the sync mark detector 30 can start searching only at certain bit positions. For example, the sync mark detector 30 may start searching at periodic bit positions. Because the sync field 14 has a repeating pattern, such as "1100", the search may be divided into four phases. If the search for a sync mark starts at any phase, it is called an all-phase detection. If the search starts only at selected phases, for example phase 4 and/or phase 2, it is called a limited-phase detection. The limited-phase detection typically produces larger d_H as compared with all-phase detection.

[0028]   Two types of errors may occur in the read back signal of a communications channel such as a disk drive. Short error events are caused by noise. As used herein, channel irregularities refers to long consecutive bit errors in the incoming bitstream. Examples of channel irregularities include thermal asperity and/or media defects. Still other channel irregularities that cause long consecutive bit errors are contemplated. Short error events usually corrupt several consecutive bits and can cause both sync mark miss and misalignment.

[0029]   Common error events are set forth in FIG. 5. "x" identifies the error that it can be either "+" or "-" depending upon the preceding bits. Since each of these error events only corrupts a small number of bits, a sync mark with a larger Hamming distance d_H can tolerate an increased number of short error events when bit-by-bit matching is performed. Because errors propagate after the $(1 \oplus D^2)$ post coding, bit-by-bit detection is preferably performed in the NRZ

domain. Channel irregularities corrupt long consecutive bits of the sync mark. Therefore, symbol-by-symbol matching is more robust when channel irregularities occur.

[0030]   The largest minimum Hamming distance d_H of any 36 bit sync mark cannot exceed 36. Therefore, 18 consecutive bit errors would fail the sync mark. However, a carefully designed 36 bit sync mark can tolerate up to 20 consecutive bit errors when symbol-by-symbol detection is used as will be described more fully below.

[0031]   Referring now to FIG. 6A, a sync mark detector 50 according to the present invention is shown. The sync mark detector 50 includes a front end 52 that provides an incoming bitstream 54 and a channel irregularity signal 56 that identifies the presence of channel irregularities. The sync mark detector 50 includes first and second operating modes. When the first operating mode is enabled by the channel irregularity signal (when channel irregularities are not present), a bit decision circuit 60 receives the incoming bitstream and generates an output to a first buffer 64. The buffer 64 is connected to a bit level comparison circuit 68 that applies a bit level matching rule 70 when comparing a sync mark 72 to the incoming bitstream.

[0032]   In the second operating mode (when channel irregularities are present), the bit decision circuit 60 is connected to an INRZI ($1 \oplus D^2$) post coding circuit 74. The INRZI post coding circuit 74 is connected to a buffer 76. The buffer 76 is connected to a symbol level comparison circuit 78 that applies a

symbol level matching rule 80 when comparing the sync mark 72 (or symbols thereof) to the coded incoming bitstream.

[0033]   For purposes of clarity, reference numbers from FIG. 6A are used in FIG. 6B to identify similar elements.   In the second operating mode (when channel irregularities are present), the bit decision circuit 60 is connected to a NRZI post coding circuit 84 (1 $\oplus$ D).   The NRZI post coding circuit 84 is connected to a buffer 86.   An output of the buffer 86 is input to a symbol level comparison circuit 88 that applies a symbol level matching rule 90 when comparing the sync mark 72 to the incoming bitstream.   As can be appreciated, the first operating mode may alternately employ NRZI or INRZI post coding. Likewise, the second operating mode may alternately employ NRZ post coding.

[0034]   The sync mark detector 50 tolerates both short and long error events.   If channel irregularities exist, symbol-by-symbol detection is provided by the second operating mode.   Otherwise, bit-by-bit detection is provided by the first operating mode.   Sync marks are specifically designed for the sync mark detector 50 using the following criteria.   Preferably "101" or "010" patterns are not allowed.   These patterns make a "+-+" error event possible.   This error pattern is a dominant error event that corrupts three bits when detected in the NRZ domain. Other common error events only corrupt 2 bits.

[0035]   In symbol-by-symbol detection, error events may occur across the symbol boundaries.   Therefore, a single error event can corrupt more than one symbol.   All of the dominant error events are less than six bits long after (1 $\oplus$ D^2) INRZI post coding except for the pattern "+000+" that propagates to seven

bits. Therefore, if each symbol contains five bits and the patterns "10001" and "01110" are forbidden to start immediately before a symbol boundary, the error events listed in FIG. 5 can only corrupt a maximum of 2 symbols. This requirement can be altered to provide various different error propagation protections. For example, if the sync mark detector 50 performs symbol-by-symbol detection in the NRZ domain, it is possible to define symbols of more than five bits and to forbid an error event to cross the symbol boundaries. In this case, an error event can only corrupt one symbol.

[0036]   For example, an exemplary 36 bit sync mark can be divided into 7 symbols with 5 bits in the first 6 symbols and 6 bits in the last symbol as shown in FIG. 7. The last symbol can be 4 bits, 5 bits or 6 bits without significantly affecting the performance of the sync mark detector. If the error event starts in the last two symbols, it can corrupt at most 2 sync mark symbols.

[0037]   If the sync mark contains m symbols, the sync mark found state is declared if at least n symbols are matched. Because the error event can potentially corrupt 2 symbols, the sync mark can tolerate [(m-n)/2] error events where [(m-n)/2] is the largest integer not exceeding (m-n)/2. Therefore, (m-n) is preferably an even number.

[0038]   In both bit-by-bit and symbol-by-symbol detection modes, the early alignment probability is preferably not greater than the probability of a sync mark miss. This requires a large minimum Hamming distance d_H for bit-by-bit detection and a large minimum symbol distance in the symbol-by-symbol detection. A good sync mark also needs to produce a high number of polarity

transitions when it is written onto a recording media or transmitted through a communications channel to ensure that the timing recovery loop contains enough information to recover the symbol clock.

[0039]   Based upon the foregoing, exemplary 64 bit, 36 bit and 34 bit sync marks were generated and are listed in Tables of the above-identified Provisional Applications. The ending sync field pattern of "1100" is taken into consideration when counting the number of transitions. As can be appreciated, other sync marks may be generated.

[0040]   The exemplary 64 bit sync mark contains 13 symbols. Each of the first 12 symbols contains 5 bits. The last symbol contains 4 bits. During symbol-by-symbol detection, the sync mark found state is declared if 5 out of 13 symbols match the predefined sync mark in the INRZI domain. Therefore, this sync mark can tolerate 4 error events during the symbol-by-symbol detection. During bit-by-bit detection, if the detector starts a sync mark search at even phases (for example, phases 2 and 4 in FIG. 4), the minimum Hamming distance d_H of the sync mark increases to 25. This sync mark can tolerate 6 error events in NRZ detection, which is particularly suited for applications without channel irregularities.

[0041]   The exemplary 36 bit sync mark contains 7 symbols. Each of the first 6 symbols contains 5 bits. The last symbol contains 6 bits. A 3 out of 7 matching rule results in a 2 error event tolerance in the symbol-by-symbol detection. During bit-by-bit detection, if the detector starts the sync mark search at even phases, a minimum Hamming distance d_H of the sync mark increases to

14. This sync mark can tolerate 3 error events during NRZ detection, which is suitable for applications without channel irregularities.

[0042]   The exemplary 34 bit sync mark also includes 7 symbols. Each of the first 6 symbols contains 5 bits. The last symbol contains 4 bits. A highly preferred pattern is "11100 00111 10001 11001 10000 11111 1110". The symbol-by-symbol detection uses a 3 out of 7 matching rule and tolerates 2 error events. The bit-by-bit detection starts searching at even phases and tolerates 3 error events with a minimum Hamming distance of 13.

[0043]   Referring now to FIG. 8, the method for operating the detector 50 is shown. In FIG. 8, it is assumed that the M-bit sync mark contains m symbols with a minimum Hamming distance d_H. The symbol-by-symbol detection requires n symbols to match. Symbol-by-symbol detection is performed in the INRZI domain with an all-phase search. Bit-by-bit detection is performed in the NRZ domain with a limited-phase search.

[0044]   Bit-by-bit detection in the NRZ domain requires the incoming bitstream to be compared with both the sync mark and its bit complement. If the bit differences in either case are less than half of the Hamming distance d_H, the detector declares a sync mark found state. A more simple method is illustrated in FIG. 8. There, the detector 50 compares the incoming bitstream with the sync mark. If the bit differences are less than half of the Hamming distance d_H, or greater than or equal to $M - [(d_H -1)/2]$, the sync mark found state is declared. In the latter case, the incoming bits are within half of the Hamming distance d_H from the inverted sync mark.

[0045]   In the sync mark detection method of FIG. 8, control begins with step 100. In step 102, the sync mark detector 50 determines whether channel irregularities exist. If channel irregularities exist, control continues with step 104 where INRZI post coding is performed. In step 106, M bits of the incoming bitstream are buffered at desired phases. In step 108, the buffer is compared symbol-by-symbol with the INRZI sync mark. In step 110, if the number of different symbols is less than (m-n), a sync mark found state is generated in step 112. Otherwise, a sync mark missed state is generated at 114.

[0046]   If channel irregularities do not exist in step 102, control continues with step 120 where M bits of the incoming bitstream are buffered at desired phases. In step 122, the buffer is compared bit-by-bit with the NRZ sync mark. In step 126, control determines whether the number of different bits is less than or equal to $[(d_H -1)/2]$ or if the number of different bits is greater than or equal to $M - [(d_H -1)/2]$. If either condition is true, control continues from step 126 to step 112. Otherwise, control continues from step 126 to step 114. Control ends in step 130. As can be appreciated, the steps set forth in FIG. 8 can be readily modified by skilled artisans to perform NRZI post processing or any combination of NRZ, INRZI, or NRZI post processing.

[0047]   Referring now to FIG. 9, the performance of the 36 bit sync mark is shown. At an input bit error rate of 10^{-5}, the failure rates for the sync mark detector 50 according to the present invention (identified at 150) is three orders of magnitude lower than conventional sync mark detection (identified at 152). If there channel irregularities do not exist (identified at 154), the difference

is increased to 8 or more orders of magnitude lower than the conventional sync mark detector.

[0048]   As can be appreciated from the foregoing, the sync mark detector according to the present invention selects the detection method according to the presence of channel irregularities. When channel irregularities exist, symbol level detection is used where each symbol contains several bits. Otherwise, bit level detection is used. The selected sync marks have large Hamming and symbol distances to guarantee a low early alignment probability when channel irregularities are both present and absent. The sync marks contain carefully designed symbol boundaries to control but not eliminate error propagation.

[0049]   Conventional sync mark designs attempt to eliminate error propagation by enforcing boundary constraints, which eliminates certain bit patterns at the boundaries and tends to increase the length of sync marks. By allowing the error events to propagate across symbol boundaries and selecting symbol lengths such that the propagation is limited to 2 symbols, the same sync mark failure rates are provided with shorter sync marks. The present invention also provides improved format efficiency in disk drive applications. In particular, if the longest error event causes x bit errors, a symbol length of (x-1) confines the error propagation to 2 symbols.

[0050]   Using the sync mark criteria set forth above, a sync mark can be readily designed with five-bit or six-bit symbols (or other lengths). In symbol detection mode when channel irregularities exist, a sync mark with (4+3s)

symbols can tolerate up to (1+s) error events by using a (2+s)-out-of-(4+3s) matching rule. For example, a 10 symbol sync mark with a 4 out of 10 matching rule can tolerate up to 3 error events.

[0051]   As can be appreciated by skilled artisans, the sync mark detector according to the present invention may be implemented in a variety of ways. For example, the sync mark detector can be implemented using discrete components, application specific integrated circuits (ASICs), a processor with memory and software, or in any other suitable manner.

[0052]   Those skilled in the art can now appreciate from the foregoing description that the broad teachings of the present invention can be implemented in a variety of forms. Therefore, while this invention has been described in connection with particular examples thereof, the true scope of the invention should not be so limited since other modifications will become apparent to the skilled practitioner upon a study of the drawings, the specification and the following claims.